REMARKS

Rejection of claims 1-14 under 35 U.S.C. §112, First paragraph

The Examiner rejected the claims for enablement based on the use of the term "processor hardware thread." The claims were amended to remove this term. Applicant respectfully requests the Examiner to reconsider the rejection under U.S.C. §112 first paragraph.

Rejection of claims 1-14 under 35 U.S.C. §112, Second paragraph

The Examiner has rejected the claims as being indefinite. Specifically, the Examiner questions how a thread can be hardware. While the term was amended to make the claim more clear, a response may be helpful. A thread is software, but the claimed invention concerns hardware multithreading. In hardware multithreading, the processor has support for executing a plurality of threads. This support is sometimes referred to in the application as a thread in the processor. Thread1 and Thread 2 (see processors 110, 112, ... 118 in Figure 1) is that portion of the processor hardware that supports hardware multithreading. The processors use the hardware thread to execute threads of software stored in memory 120 (page 10, lines 5-6). In context of the specification and the drawings, the claims are not indefinite and would be understood by one of ordinary skill in the art. Applicant respectfully requests the Examiner to reconsider the rejection under U.S.C. §112 second paragraph.

Rejection of claims 1-14 under 35 U.S.C. §103(a)

The Examiner rejected claims 1-14 under 35 U.S.C. §103(a) as being unpatentable over the combination of Chastain and Brenner. Applicant traverses the Examiner's finding of obviousness and believes the cited art, singularly or in combination does not teach or suggest the claimed invention herein as amended herein.

Claim 1

Chastain teaches a multiprocessor system that executes a parallelization instruction over multiple processors. Brenner teaches load balancing in a multiple run queue system. In Brenner, the system is a software multithreaded system, meaning the support for multithreading is done with software using software supported queues. The hardware multithreaded system claimed herein does not have software run queues as described in Brenner. Neither Brenner or Chastain even deal with hardware multithreading.

Claims 1 was amended to add the limitation that the thread dispatch mechanism determines which of the processors is busy processing a thread but can accept an additional thread. Brenner combined with Chastain does not teach or suggest to determine when a processor is busy but can accept an additional thread. The combination of the cited art does not teach or suggest to distinguish between a processor that is busy and cannot accept an additional thread and one that is busy but can accept an additional thread.

For the claim limitation "determining which of the plurality of processors are idle, which of the plurality of processors can accept an additional thread, and which of the plurality of processors cannot accept an additional thread since it is working on a maximum number of threads the processor can execute," the Examiner cited Brenner, col.

4 lines 47-50, 54-56, col. 5 lines 37-45 and col 10 lines 49-65. Brenner does teach determining when a processor is idle to perform load balancing (col. 4, line 49). But Brenner does not teach or suggest to differentiate between a processor that is idle, one that is busy but can accept an additional thread, and one that cannot accept an additional thread. In Brenner, there is no hardware support for hardware multithreading, so there are only two states of a processor: idle, meaning the processor is doing nothing; or busy, meaning the processor cannot accept an additional thread. In the amended claims, the dispatch mechanism determines if a processor is busy, whether it is busy but can accept an additional thread, or whether it can't accept an additional thread. Brenner does not teach or suggest to make this distinction over three possible states of the processor.

Since Brenner and Chastain do not teach or suggest to determine whether a processor is busy but can accept an additional thread, claim 1 is allowable over the combination of Brenner and Chastain. Applicant respectfully requests reconsideration of the rejection of claim 1 under 35 U.S.C. §103(a).

Claim 2

Claim 2 depends on claim 1, which is allowable for the reasons given above. As a result, claim 2 is allowable as depending on an allowable independent claim. Further, claim 2 contains an additional claim limitation that is not taught or suggested by the cited art. For claim 2 the Examiner cited Brenner, col. 4 lines 45-56; col. 5, lines 37-45 and col. 10 lines 49-65. Applicant has not found anything in the cited section, or in Brenner in general to support the Examiner's rejection. Column 4, lines 45-56 describes placing a thread in a run queue of an idle CPU. Column 5, lines 37-45 describes load balancing threads to a next available CPU. Column 10, lines 49-65 describes load balancing to a lightest loaded CPU run queue. In each case, Brenner is concerned with software multithreading techniques. The cited passages deal with assigning threads to a software queue depending on whether a processor is idle or not. The cited sections of Brenner do not teach or suggest “if none of the plurality of processors is idle and if at least one of the plurality of processors can accept an additional thread, the thread dispatch mechanism dispatches the new thread to one of the plurality of processors that can accept an additional thread.” Brenner does not address whether a processor can accept an additional thread. In Brenner, the threads are dispatched depending on the idle status only, and the threads are dispatched to a run queue and not to the processor. Applicant respectfully requests the Examiner to reconsider the rejection of claim 2 under 35 U.S.C. §103(a).

Claim 3

Claim 3 depends on claim 1, which is allowable for the reasons given above. As a result, claim 3 is allowable as depending on an allowable independent claim. Further, with regards to the rejection of claim 3, the Examiner states that Brenner discloses the limitation of “the thread dispatch mechanism waits for one of the plurality of processors to complete processing” recited in claim 1, citing Brenner at col. 5 lines 37-45, and col. 10 lines 49-65. Applicant has not found this teaching in the referenced portions, or any other portion of Brenner. In Brenner, a thread is assigned to an idle thread or the “lightest loaded run queue” (col. 10, line 41). There is no discussion in Brenner concerning waiting for a processor to become a processor that can accept a thread. Since Brenner and Chastain do not teach or suggest to wait until a processor can accept an additional thread, Applicant respectfully requests reconsideration of the rejection of claim 3 under 35 U.S.C. §103(a).

Claims 4-11

Claim 4 and 7 contain similar limitations to those described above with reference to claim 1. Therefore claims 4 and 7 are also allowable over the cited art. Further, claims 5-6 and 8-11 depend on claims 4 and 7 respectively, which are allowable for the reasons given above. As a result, claims 5-6 and 8-11 are allowable as depending on allowable independent claims.

Claims 12-14

Claims 12-14 depend on claims 1, 4 and 7 respectively, which are allowable for the reasons given above. As a result, claims 12-14 are allowable as depending on allowable independent claims.

Conclusion

In summary, none of the cited prior art, either alone or in combination, teach, support, or suggest the unique combination of features in applicant's claims presently on file. Therefore, applicant respectfully asserts that all of applicant's claims are allowable. Such allowance at an early date is respectfully requested. The Examiner is invited to telephone the undersigned if this would in any way advance the prosecution of this case.

Respectfully submitted,

By /bretjpetersen/
Bret J. Petersen
Reg. No. 37,417

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700